Internal Version (Early Draft)

Chaining Geoinformation Services for the Visualization and Annotation of 3D Geovirtual Environments

Benjamin Hagedorn, Stefan Maass, Jürgen Döllner

Hasso Plattner-Institute – University of Potsdam,
Prof. Dr. Helmert Str. 2-3, 14482 Potsdam, Germany
Phone ++49 331 97992 580, Fax ++49 331 97992 589
{benjamin.hagedorn, stefan.maass, doellner}@hpi.uni-potsdam.de

Abstract: Visualization of 3D geovirtual environments enables users to gain insight into and to interactively operate with complex 3D geoinformation as well as to explore and analyze underlying structures, relationships, and associated thematic georeferenced data. It represents a key functionality for a growing number of applications and systems such as for way-finding, urban city planning, geo-marketing, and tourist information. For the systematic implementation of these systems, geoinformation services provide an interoperable and standardized way to access, process, and view distributed, heterogeneous 2D and 3D geospatial data. To broaden the use and scope of 3D geovirtual environments, we have to support the management and visualization of user-defined or application-defined data associated with geospatial objects. For this purpose, we can embed annotations such as texts or symbols as meta elements into images of 3D geovirtual environments. For example, annotations can represent georeferenced comments added interactively by the user to explicitly denote spatial objects and locations the comments refer to. This paper presents the design of a chain of web services for the creation of annotated perspective views. First, a Web Perspective View Service (WPVS) is outlined that synthesizes images of a given 3D geovirtual environment and delivers the images to the service consumer. Second, the Web View Annotation Service (WVAS) is introduced that processes these images as input along with a set of spatial locations and content definitions. To ensure high legibility, annotations are embedded into the view plane and placed such that they avoid occlusions among themselves and with geospatial objects of the scene. As result of this service composition, the service consumer receives an enhanced image depicting the current perspective view with seamlessly embedded annotations. We describe the design and system architecture by the example of a prototypic interactive campus information system.

Keywords: 3D Geovirtual Environment, Geovisualization, Annotation, Geoinformation Services, Service Composition.

1 Introduction

1.1 3D Geovirtual Environments

3D geovirtual environments (3D GeoVE) provide a conceptual and technical framework for the seamless integration of geoinformation that is different in format, scale, and the amount of details. They provide mechanisms to enable the composition, management, editing, analysis, and visualization of this integrated geoinformation. 3D city models and 3D landscape models are examples for such 3D GeoVEs that can be composed out of different geoinformation models such as a terrain, building, vegetation, street network models, and additional thematic data. The visualization pipeline (Spence, 2001) of 3D GeoVEs includes accessing and selecting geodata, mapping them to computer graphical representations, and rendering of depictions, which can be perceived by humans. The following issues are relevant for that geovisualization process:

- *Data selection* describes the process of filtering the available geodata according to the application and problem domain.

- *Styles* describe appearance parameters to be applied to the visualization of the selected geoinformation.
- *Rendering parameters* describe attributes that influence the image synthesis process, e.g., image size, image resolution, or color schemes adapted to special output devices or user groups.
- *Interaction properties* describe the possibilities of manipulating the 3D GeoVE (e.g., navigating, selecting, and editing) and take different input devices into account.

1.2 Annotations

Annotations are essential elements to communicate textual and symbolic information for cartographic maps and within 3D GeoVEs. Traditionally, they name point, line, or area features, such as locations, streets, rivers, districts, or lakes. A number of criteria, e.g., a clear correlation with the feature, the legibility of the annotation, and the occlusion of other annotations or important image parts have to be considered to optimize the information transfer to the user and to achieve an aesthetical appearance of the annotated depiction. The use of electronic media for map creation and presentation raises the need for automated annotation techniques. This need has been strengthened by the increased use of interactivity in today's geovisualization applications and systems, which allows users the real-time exploration, analysis, and modification of geospatial data.

Automated annotation techniques have been developed for map production first. An overview of techniques for point feature label placement on maps is given by Christensen et al. (1995). Techniques that can label line and area features are presented by Edmonson et al. (1997) and Wolff et al. (1999). The more recent approaches of Petzold et al. (2003) and Been et al. (2006) explicitly focus on dynamic labeling during zooming and panning of 2D maps on computer screens.

A first approach for the automated labeling of interactive 3D GeoVEs was presented by Bell et al. (2001). They developed a view management data structure that efficiently supports the registration and query of rectangles on the view plane. This is used to mark such regions as occupied that show important scene elements or formerly placed labels. Maass and Döllner (2006b) present another view management strategy that is optimized for point feature labeling of terrains. Furthermore, they develop a technique that directly embeds annotations as 3D elements into the 3D scene instead of presenting them as screen overlays (Maass and Döllner 2006a, 2007).

1.3 Service-Based Geovisualization

Geoinformation resources and capabilities for geodata processing and geovisualization are increasingly distributed among various providers. Service-oriented computing is concerned with describing, finding, and using these capabilities for different processes, users, and tasks. In the field of distributed geoinformation systems we can distinguish between geodata access, geodata processing, and geodata visualization services. The Open Geospatial Consortium (OGC) has standardized a set of geoinformation services that define interfaces to these functionalities. Important OGC web service standards are the Web Feature Service (WFS) for geodata access, the Web Map Service (WMS) for 2D maps, and the Geography Markup Language (GML) for the interoperable description of geospatial and georeferenced data. The Web Perspective View Service (WPVS), a first approach for 3D portrayal, is currently under discussion to become an OGC standard. According to the separation of the visualization process between service provider and service consumer as described by Altmaier and Kolbe (2003), a WPVS represents

a fat server that implements the whole image synthesis process. It produces 2D images showing perspective views of a 3D GeoVE, which are transferred to and presented by thin clients such as Smartphone or PDA. As one example, the open source platform deegree contains an implementation of a WPVS (deegree 2007).
For the specification of rendering styles the OGC supports the Styled Layer Descriptor (SLD) and Symbology Encoding (SE) specifications (Lalonde 2002, Müller 2006). Together with a reference to the input data, a WMS consumer can use predefined styles for the geovisualization or specify own styles to be applied. Annotations can be integrated into the WMS-based geovisualization in two ways: The SLD and SE specifications support textual and graphical annotations by TextSymbolizers (for label placement) and PointSymbolizers (for 2D graphics placement). Until now, for the styling of 3D GeoVEs, no standards emerged.

1.4 Service Composition

A core concept of service-based systems represents the composition of distributed functionality in a standardized manner. This enables the construction and flexible adaptation of complex and value-added systems and applications. In the geoinformation domain, service composition is often referred to as *geoinformation service chaining*. Alameh (2003) distinguishes three service composition patters: Client-based chaining, aggregate services, and workflow service-based chaining. Neis et al. suggest an Accessibility Analysis Service (AAS, Neis et al. 2007a) and an Emergency Route Service (Neis et al. 2007b), which employ service composition by aggregate services. For example, they combine street network data from WFS with processing capabilities of their AAS (which generates new geometry described as GML) and WMS mapping capabilities. Weiser et al. (2007) show the possibility of using BPEL engines together with WSDL service descriptions of OGC web services for a workflow-based service orchestration.

1.5 Challenges for Implementing Annotation Techniques as Web Service

In our approach, we concentrate on combining scene rendering for 3D GeoVE and the automated annotation of contained 3D objects. We address the following questions:

- *Design Constraints:* Which adaptations to an application implementation are required to provide the functionality of the annotation technique as a web service? Does the service concept restrict the functionality?
- *Data Specification and Web Service Chaining:* How should the interface of the annotation service be designed to achieve a high interoperability and reusability?
- *Support of User Interaction:* How can the interactive modification of annotations and viewing parameters be supported when implemented as a web service chain?

A strong motivation for our work was the assumption that answering these questions for the service-based annotation of 3D GeoVEs helps to extend the capabilities of current service-based geovisualization systems, which includes the utilization of sophisticated visualization techniques and the support of task-oriented and user-oriented interaction.

2 Concept for Service-Based 3D View Annotation

Our approach to annotated 3D views of GeoVEs combines two main services, an extended Web Perspective View Service (WPVS) and the Web View Annotation Service (WVAS). The service chain is implemented as client-based service composition. Complementary to the thick geovisualization service, this client is constructed as a thin client. It knows about the component

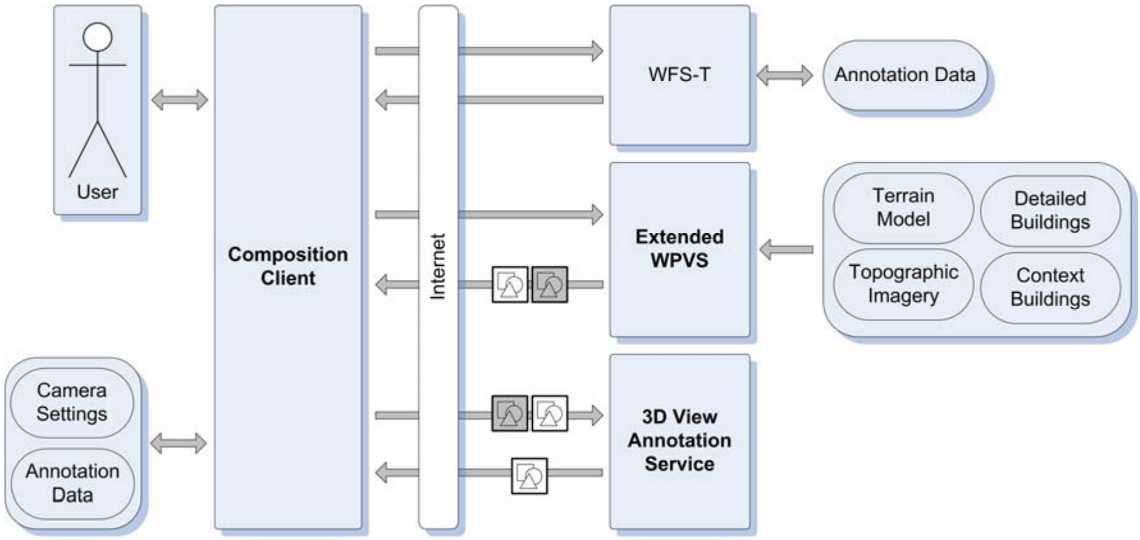

Figure 1: Architecture of the service chain for annotated views of GeoVEs.

services and completely controls the service-based geovisualization process. Figure 1 illustrates the overall architecture of our implementation and the main message transfer between the involved components.

Our approach to an extended WPVS encapsulates the access to and the integration of geoinformation in one 3D GeoVE comprising large terrain models, large aerial images, and building data in different formats (e.g., CityGML, 3DS). It is capable of synthesizing high-quality images using rendering techniques for ambient occlusion and atmospheric effects such as sun and clouds. In our case study, the service provides access to the integrated GeoVE of a 3D campus model composed of a terrain model, topographic imagery, and several building models such as main building, auditorium, library, cafeteria, and nearby train station. The composition client receives a perspective view and a depth image from the extended WPVS and forwards them to the WVAS along with a set of annotation descriptions and configurations. Finally, the WVAS calculates and overlays the embedded textual annotations.

Currently, the client application is implemented on a JavaScript basis. As with AJAX, this enables asynchronous communication with web services using the XMLHttpRequest object with SOAP messages. The input and output image data is encoded as base64-string and transmitted as part of the SOAP request or response messages, respectively. Figure 2 shows a screenshot of the client. It enables simple navigation, allows the user to define own annotations and to request annotated views.

The client application provides the following main functionality:

- *Adapter to the Web Perspective View Service:* The client application is capable of composing a request to the WPVS and extract image data from the response.
- *Adapter to the Web View Annotation Service:* The perspective view is forwarded to the 3D WVAS together with annotation definitions. The client stores the annotations to embed into the perspective view.
- *Image portrayal:* The synthesized and enriched perspective view of the 3D GeoVE is presented to the user as image.

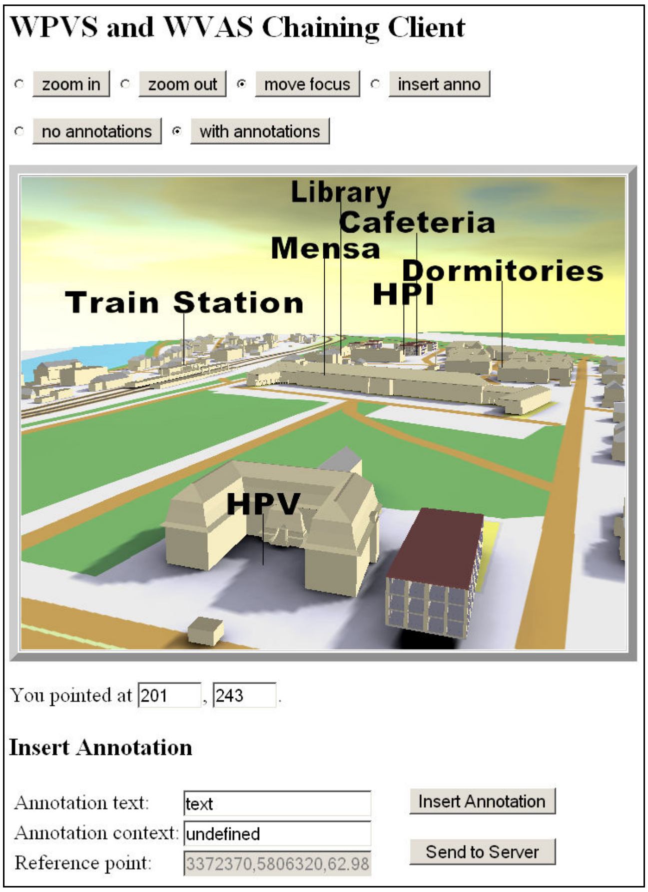

Figure 2: Screenshot from our thin client showing an overview of the campus area.

- *Interaction support:* The client application allows the user to navigate in and edit the 3D GeoVE represented by the displayed perspective view. For this, it tracks the camera parameters (e.g., camera position and look-to direction) and mouse interactions with the image and with elements of the user interface (buttons).

For adding user-defined annotation to the 3D GeoVE, we have integrated a transactional WFS in our WPVS. This data can be requested by other users and can be utilized for creating the WVAS request.

Functionality and interface of the WPVS have been extended to enable their composition with the WVAS. Both services are described in Section 3 and Section 4.

3 The Web View Annotation Service

3.1 Annotation Technique for Point-Features

In contrast to traditional paper maps, there is no single annotation technique for 3D GeoVEs that would fit all needs of all users in all application contexts. For example, in way-finding applications using the virtual pedestrian metaphor, users are only interested in a few, properly embedded annotations that help to make a decision for the current viewing position. In contrast,

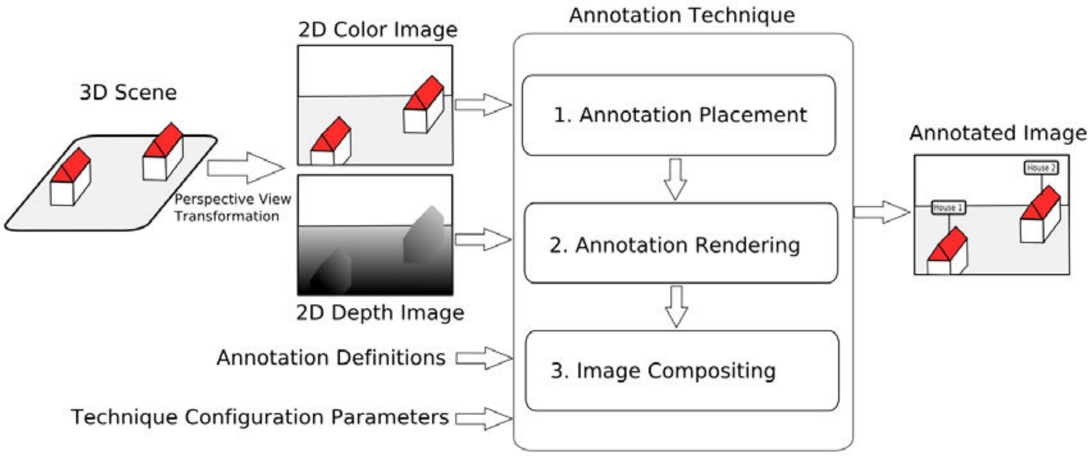

Figure 3: Overall process for the annotation of 3D scenes.

the interactive analysis of large-scale geographical areas, where users take a bird's eye perspective, calls for techniques that support large numbers of annotations that are all visible at once.

For our web service implementation, we have selected the annotation technique for point features in 3D terrain-based scenes described by Maass and Döllner (2006b) for two reasons. First, it is suitable for a wide range of application contexts and, second, a working application-integrated implementation exists. To make it usable for service chaining, we had to modify the original technique slightly such that scene elements and annotations are rendered in a separate rendering pass whose results are combined with the scene depiction in a final rendering pass. The visual results, however, remain the same.

Figure 3 illustrates the rendering process including input and output of the annotation component. Starting with a 3D scene description, a perspective view transformation generates an RGB color image and a depth image that encodes the depth of the visible scene elements to the observer with values linearized between *0* (in front of or on the camera near plane) and *1* (on or behind the camera far plane). Both images together with annotation positions and texts, as well as a set of configuration parameters (e.g., font size, color, annotation style) serve as input to the annotation technique. It calculates the positions of the labels, renders them into a separate image buffer, and combines the resulting image in a depth-sensitive way with the input color image.

3.2 Architecture and Implementation of the WVAS

The WVAS provides the single operation `GetAnnotatedView`. Service request and response are encoded as SOAP messages sent over Http. For both, input and output, images can be transferred in different ways. First, a URL pointing to the image on a web server or representing a URL-encoded service request, e.g., to a WPVS, can be used. Second, the images can be submitted within or attached to the SOAP message as described by Powell (2004), e.g., using the SOAP extension DIME (Direct Internet Message Encapsulation) for attaching binary data. In our current implementation the color and depth images are encoded as base64 strings and sent within the SOAP messages.

The WVAS is implemented as a stateless service: It does not provide any user or session handling and does not store any camera, canvas, or annotation descriptions. Therefore all input data has to be included in every service request. Some of the data that defined the preceding image rendering process has to be included into the service request as well. In detail the GetAnnotatedView operation uses the following input parameters:

- *List of annotations:* An annotation is defined by an annotation text and a 3D position, the annotation's reference point. The annotation can be described as an abstract GML feature containing a georeferenced gml:Point (see Listing 1 for an example).
- *2D color image:* The 2D color image represents a perspective view of a 3D scene for a specific viewpoint and contains RGB values for each pixel.
- *2D depth image:* The 2D depth image is related to the color image. Each pixel stores the distance of the visible scene element to the camera with float precision.
- *Camera definition:* As usual in the 3D computer graphics domain, the camera is defined by the look-from vector, look-to vector, look-up vector, near plane, far plane, and field-of-view angle. This is different from the camera model of the OGC WPVS which is defined by the point of interest (that is the look-to point in our model), the camera distance to that point, angles describing the north direction, pitch, and field-of-view. However, both models can be transformed into each other.
- *Canvas definition:* The canvas definition describes the width and height of the input color and depth images.
- *Annotation configuration:* The appearance of the annotations generated by the technique can be adjusted by parameters such as the placement variant, color, font, and annotation size.

Listing 1: Example of an annotation specification.

```
<Annotation>
    <text>Hasso-Plattner-Institute</text>
    <position>
        <gml:Point srsDimension="3" srsName="EPSG:25833">
            <gml:pos>372884.118 41.0 5806445.085</gml:pos>
        </gml:Point>
    </position>
</Annotation >
```

The WVAS is currently implemented as .NET Web Service executed by the Microsoft Internet Information Services (IIS). Its implementation is based on the Virtual Rendering System, "a computer graphics software library for constructing interactive 3D applications. It provides a large collection of 3D rendering components […]." (VRS 2007, Döllner and Hinrichs 1995)

4 The Extended Web Perspective View Service

This section describes the WPVS used to render images of complex 3D GeoVEs. In particular, it supports rendering of large-scale digital terrain models, aerial images, terrain textures defined by 2D maps, and 3D site models including buildings and infrastructure elements. Perspective views of the 3D GeoVE can be requested using the GetView operation with camera definition and image dimensions. The WPVS has been implemented on top of the LandXplorer, a system and framework for the management and real-time visualization of complex virtual 3D city and landscape models (Döllner et al. 2003).

4.1 Extension for Depth Image Provision

The WPVS has been extended by providing additional image-encoded scene information, i.e., the service not only delivers the RGB image of a 3D GeoVE but also an additional image that encodes scene depth. The depth image cannot be created by SLD or SE feature visualization styles, which only influence the mapping of geoinformation to graphical primitives. Instead, the rendering stage in the visualization pipeline has to be modified. We identified at least the following possibilities for integrating such rendering functionality into WPVS:

- *Extending the GetView operation by an additional RenderingStyle parameter:* Depending on this parameter the service decides about the creation of the default color image or the depth image and the format in which they are delivered (e.g., as application/octet-stream for raw binary data).
- *Extending the WPVS service interface by an additional operation GetDepthView:* This operation generates the depth image and only provides appropriate transfer data. The further parameters of the operation are identical to the GetView request.

Both ways provide access to intermediate data of the visualization pipeline, which have not been accessible before to service consumers. This data is not intended for perception by humans but serves as input for new service-based visualization techniques.

Our WPVS implements the additional operation GetDepthView. Nevertheless, the option of explicitly supporting RenderingStyles seems to be very promising as it is a more general concept for enabling additional rendering techniques.

4.2 Service-Based User-Interactivity

To support user interaction, we have extended the WPVS by a GetFeatureInfo operation. Corresponding to the WMS GetFeatureInfo operation, it provides additional information about features in the perspective view of the 3D GeoVE that is returned by a previous GetView request. Because of the stateless implementation of the WPVS the GetFeatureInfo request has to specify most of the parameters of the GetView request, i.e., canvas and camera settings, layers to include, and styles to apply. As further parameter the 2D image position of interest is specified.

The server-side implementation of GetFeatureInfo uses a ray intersection test for identifying the selected objects. A 3D ray request is originated at the camera position and shot into the scene. The primarily hit GeoVE object is evaluated and thematic information such as the GeoVE object type (e.g., building, roof, terrain, etc.), object identifier, and geospatial position can be derived and included into the GetFeatureInfo response message.

For the interactive specification of annotations, this position can be used as annotation anchor point. Additionally, object-related information such as the object center, footprint center, or other predefined building-related points of interest (e.g., meeting places, elevators, emergency exits) can be delivered to support the user in specifying annotations.

5 Results

We have successfully implemented a client-controlled service-chain supporting the 3D annotation of 3D GeoVEs. For example, Figure 4 shows two annotated views of the

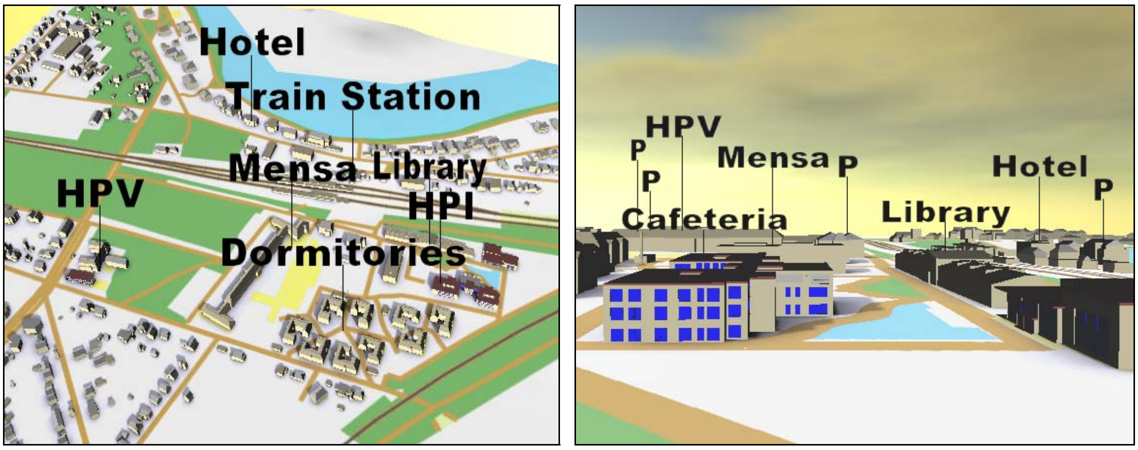

Figure 4: Screenshots of annotated views of the 3D campus model from a bird's eye (left) and close to ground (right) perspective generated by the service chain implementation.

3D GeoVE of our campus and how the automatic annotation supports a high legibility by preventing occlusions with scene objects and other annotations.
The functionality and interface of the WPVS were extended to enable the generation and delivery of depth images. The WVAS uses offline rendering to generate the image that contains the rasterized annotations.
To enable user interactivity in applications and systems that use the described service chain, the WPVS has been extended by the GetFeatureInfo operation. Complementary, the composition client has to catch the user input and forward it to the extended WPVS. This is the same for the user's navigation input, which is mapped to a new camera position and results in a new execution of the service chain.

Additionally, session state handling is addressed, including tracking the camera settings and determining the annotations to be added to the perspective view. In our case, the session handling was performed by the composition client itself. Server-side session handling is another approach, which could be implemented with a service-based composition approach. The sequence diagram in Figure 5 illustrates the steps performed by the composition client to process the user input (for annotation definition) and combine the extended WPVS and WVAS (for annotated view creation). This scenario contains the storage of user-defined annotations at the WFS-T.

Compared to the previous application-integrated implementation, the service-based annotation does not reach interactive frame rates. It takes several seconds until the annotated view is available for portrayal by the client. For that, we see two main reasons: First, the image data is currently handed over five times between the composition client and the component services. This could be reduced by using a web server as a storage place for this data and only transferring the image URL to the WVAS. Second, the images are currently base64-encoded what boosts the data size by a third. Here, other technologies for sending binary data (even streaming them) would reduce the network load.

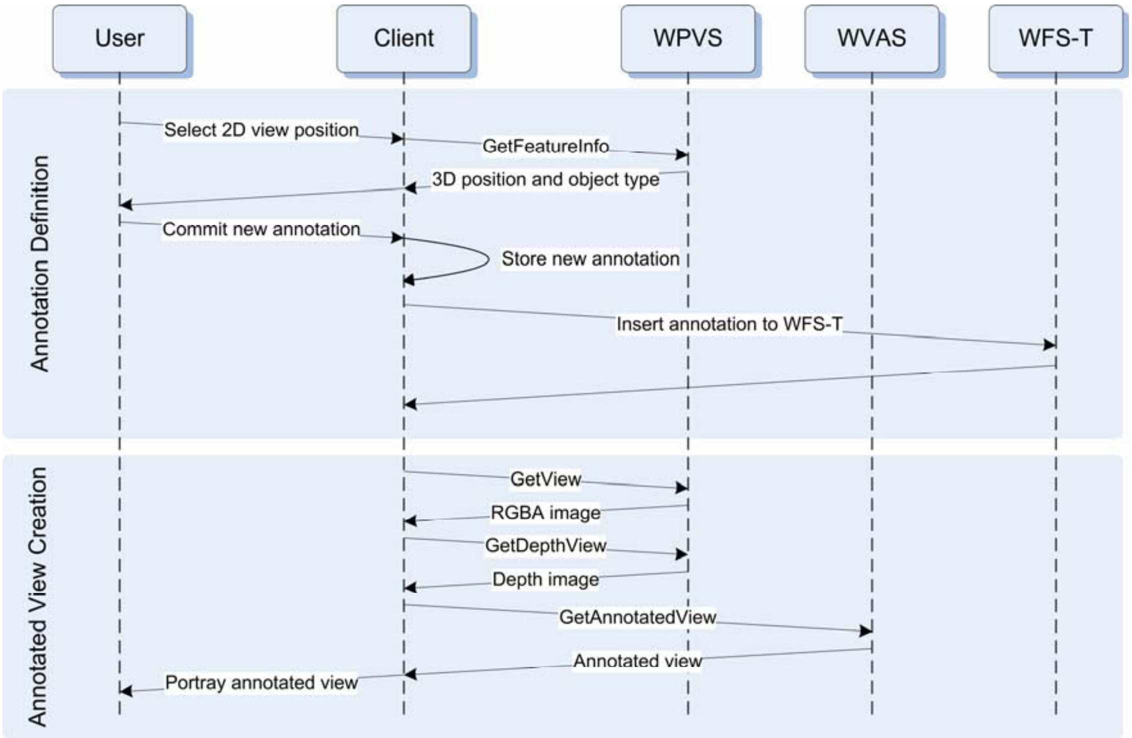

Figure 5: Sequence diagram describing the overall workflow for defining and creating the annotated perspective view.

6 Conclusions

The presented concept and prototype show how two complementary 3D visualization techniques can be seamlessly combined by implementing these techniques by two independently designed, implemented, and deployed web services; chained together they form a higher-level web service chain. In particular, separating core scene rendering of 3D GeoVE from specialized 3D visualization techniques, for example annotation rendering, facilitates the systematic, modular composition of complex visualization applications and simplifies their implementation. For example, the annotation service could be reused in other web service chains, or it could be enhanced by a navigation information service.

The presented concept can be applied to all visualization services that basically operate in image space provided that the WPVS offers additional scene information. This way, we can use an optimized 3D rendering engine for the most time-critical part, the rendering of complex 3D GeoVEs, and other aligned web visualization services do not depend on the underlying internal representation – they rely on a clearly defined, image-based interface. This approach also offers a high degree of interoperability because the individual web services do not need to exchange contents of 3D GeoVE, for which no commonly accepted standards exist so far. Furthermore, in the case of large-scale, content rich 3D GeoVE, exchanging this content through a chain of web services would be non-optimal unless these services would actually modify the content.

The extensions to the WPVS can be generalized: In all standard 3D graphics systems (e.g., OpenGL, DirectX), the provision of image-encoded scene information (e.g., depth, surface normals, object identities, etc.) represents a common feature. Therefore, we suggest including

these extension into the official WPVS definition. For example, higher-level web services that provide advanced stylized images (e.g., illustrations) could be implemented this way.
Due to the decomposition, a communication overhead results between the chained components. Since we can assume that the presented web service chain is not intended to implement a real-time interactive 3D application but to provide a perspective view service with reduced requirements regarding interactivity, the overhead can be accepted.

In our future work, we will address the following aspects. First, we will implement the service chain by an aggregate service, which includes the logic for service chaining, currently implemented by the composition client. Second, we will provide web-service adapters to general web services (WMS, WFS), which provide access to geodata (e.g., maps, 3D terrains, and city models), offer lookup services whose results can be visualized by annotations (e.g., taxi stations, restaurants, or parcel tracking information), and use the annotation service for analysis and exploration tasks. Third, we will implement additional annotation techniques. Here, we want to investigate techniques that integrate annotations into 3D scene descriptions. Fourth, we want to enhance user interaction implemented by web services. This includes functionality to interactively specify annotations also for line and area features, used to add user comments in urban city planning scenarios, as well as to support network-based multi-user collaboration.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the InnoProfile research group “3D Geoinformation” (www.3dgi.de). We also would like to thank 3D Geo GmbH, Potsdam (www.3dgeo.de) for their cooperation in developing the LandXplorer geovisualization platform.

References

Alameh, N. Chaining Geographic Information Web Services. In Internet Computing, IEEE Computer Society, Sept.-Oct. 2003, pp. 22-29, 2003.

Altmaier, A. and Kolbe, Th. Applications and Solutions for Interoperable 3D Geo-Visualization. In: Fritsch, D. (ed.): Proceedings of the Photogrammetric Week 2003 in Stuttgart, Wichmann Verlag, Stuttgart, 2003.

Bell, B., Feiner, S., and Höllerer, T. View Management for Virtual and Augmented Reality. In Proceedings of the 14th ACM Symposium on User Interface Software and Technology (UIST), ACM Press, pp. 101-110, 2001.

Been, K., Daiches, E., and Yap, C. Dynamic Map Labeling. IEEE Transactions on Visualization and Computer Graphics 12(5), pp. 773-780, 2006.

Christensen, J., Marks, J., and Shieber, S. An Empirical Study of Algorithms for Point-Feature Label Placement. ACM Transactions on Graphics, 14(3), pp. 203-232, 1995.

Deegree Software Framework. http://www.deegree.org. 2007.

Döllner, J., Baumann, K., and Kersting, O. LandExplorer – Ein System für interaktive 3D-Karten. In Kartographische Schriften, Vol. 7, pp. 67-76, 2003.

Döllner, J. and Hinrichs, K. The Virtual Rendering System - A Toolkit for Object-Oriented 3D Rendering. EduGraphics - CompuGraphics Combined Proceedings, pp. 309-318, 1995.

Edmondson, S., Christensen, J., Marks, J., and Shieber, S. M. A General Cartographic Labeling Algorithm. In Cartographica 33(4), pp. 13-23, 1997.

Lalonde, W. (ed.) Styled Layer Descriptor Implementation Specification. Open Geospatial Consortium. September 2002.

Maass, S. and Döllner, J. Dynamic Annotation of Interactive Environments using Object-Integrated Billboards, In Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'2006), Plzen, Czech Republic, pp. 327-334, 2006a.

Maass, S. and Döllner, J. Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes. In Proceedings of the 6th Int. Symposium on Smart Graphics 2006, Vancouver, Canada, pp. 1-12, 2006b.

Maass, S. and Döllner, J. Embedded Labels for Line Features in Interactive 3D Virtual Environments. In Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualization and Interaction in Africa (ACM AFRIGRAPH 2007), October 2007 (to appear).

Müller, M. (ed.) Symbology Encoding Implementation Specification. Open Geospatial Consortium, July 2006.

Neis, P., Dietze, L., and Zipf, A. A Web Accessibility Analysis Service Based on the Open LS Route Service. 10th AGILE Internation Conference on Geographic Information Science. Aalborg, Denmark, 2007a.

Neis, P., Schilling, A., and Zipf, A. Interoperable 3D Emergency Routing Based on OpenLS. 3rd International Symposium on Geoinformation for Disaster Management (GI4DM), Toronto, Canada, 2007b.

Petzold, I., Gröger, G., and Plümer, L. Fast Screen Map Labeling – Data Structures and Algorithms. In Proceedings of the International Cartographic Conference (ICC’03), Durban, South Africa, 2003.

Powell, M. Web Services, Opaque Data, and the Attachments Problem. Web Services Technical Articles. The Microsoft Developer Network Library. http://msdn2.microsoft.com/en-us/library/ms996462.aspx, 2004.

Spence, R. Information Visualization, Addison Wesley, 2001.

Weiser, A. and Zipf, A. Web Service Orchestration of OGC Web Services for Disaster Management. 3rd International Symposium on Geoinformation for Disaster Management (GI4DM), Toronto, Canada, 2007.

Wolff, A., Knipping, L., van Kreveld, M., Strijk, T., and Agarwal, P. K. A Simple and Efficient Algorithm for High-Quality Line Labeling. In Martin, D. and Wu, F. (eds.) Proc. GIS Research UK 7th Annual Conference (GISRUK’99), Southampton, pp. 146-150, 1999.

VRS. The Virtual Rendering System, http://www.vrs3d.org, 2007.